Durham University

Durham Research Online

Deposited in DRO:

22 November 2019

Version of attached file:

Published Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Erickson, A. and Navaridas, J. and Stewart, I.A. (2020) 'Relating the bisection width of dual-port, server-centric datacenter networks and the solution of edge-isoperimetric problems in graphs.', Journal of computer and system sciences., 108 . pp. 10-28.

Further information on publisher's website:

https://doi.org/10.1016/j.jcss.2019.08.005

Publisher's copyright statement:

© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full DRO policy for further details.

Durham University Library, Stockton Road, Durham DH1 3LY, United Kingdom
Tel : +44 (0)191 334 3042 | Fax : +44 (0)191 334 2971
https://dro.dur.ac.uk

Journal of Computer and System Sciences 108 (2020) 10–28

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Relating the bisection width of dual-port, server-centric datacenter networks and the solution of edge isoperimetric problems in graphs

Alejandro Erickson [a,1], Javier Navaridas [b,2], Iain A. Stewart [a,1]

[a] School of Engineering and Computing Sciences, Durham University, Science Labs, South Road Durham DH1 3LE, UK
[b] School of Computer Science, University of Manchester, Oxford Road, Manchester M14 9LP, UK

ARTICLE INFO

Article history:
Received 27 February 2017
Received in revised form 10 June 2019
Accepted 15 August 2019
Available online 27 August 2019

Keywords:
Datacenter networks
Server-centric datacenter networks
Stellar datacenter networks
Isoperimetric problems
Bisection width
S-bisection width

ABSTRACT

Stellar datacenter networks are a recent generic construction designed to transform a base-graph into a dual-port, server-centric datacenter network. We prove that the *S*-bisection width of any stellar datacenter network can be obtained from the solution of isoperimetric problems on the base-graph, provided that the base-graph is regular. We extend previous research on the stellar datacenter networks GQ*, instantiated with generalized hypercubes, and show that with respect to *S*-bisection width, GQ* performs well in comparison with the dual-port datacenter network FiConn. Our work develops a strong combinatorial link between graph bisection width and throughput metrics for stellar datacenter networks.
© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The design of datacenter networks is becoming an important aspect of computing provision as software and infrastructure services increasingly migrate to the cloud. The actual networks themselves have evolved as computational demands have increased, and various new paradigms have emerged as regards the next generation of datacenter networks. A notable evolutionary shift was provoked by Al-Fares et al. [3] when they moved away from the hitherto tree-based datacenter networks through the use of fat-trees, which originated with Leiserson [30] and are used in supercomputer design. The resulting datacenter network Fat-Tree is a switch-centric datacenter network whereby all communication intelligence resides in the (high-end) switches. This is symptomatic of a general phenomenon: the vast amount of research on interconnection networks (designed for networks-on-chips, distributed-memory multiprocessors, clusters, and so on) provides a source of ideas for both datacenter network topologies and solutions of datacenter network design problems. Of course (and germane to this paper), the differing demands and constraints of datacenter networks mean that these ideas are not always immediately applicable.

E-mail addresses: alejandro.erickson@gmail.com (A. Erickson), javier.navaridas@manchester.ac.uk (J. Navaridas), i.a.stewart@durham.ac.uk (I.A. Stewart).
[1] Supported by EPSRC grant EP/K015680/1: 'Interconnection Networks: Practice unites with Theory (INPUT)'.
[2] Supported by EPSRC grant EP/K015699/1: 'Interconnection Networks: Practice unites with Theory (INPUT)' and by the European Union's Horizon 2020 programme under grant agreement No. 671553 'ExaNeSt'.

https://doi.org/10.1016/j.jcss.2019.08.005
0022-0000/© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Whilst fat-trees are a step forward, they are not a panacea; for example, fat-trees deliver high bandwidth but are difficult to scale. Other paradigms have been proposed including: similar switch-centric datacenter networks but based upon Clos networks, like VL2 [18]; unstructured datacenter networks incorporating random graph topologies, like Jellyfish [38]; highly connected clique-based topologies such as Dragonfly [27]; hybrid solutions that route certain structured communications through a high-speed optical switch, like Helios [16]; and even mutable topologies that are interconnected by free-space optics, like Firefly [23]. Although not necessarily immediately deployable, such proposals and ideas form a valuable part of the research literature: they might either become practical in the future (or, at least, variations might); and they might inspire analytical results that can be reused elsewhere or lead to more pragmatic designs.

Another type of datacenter network has recently been proposed whereby the servers not only undertake the intrinsic computations but also deal with all aspects of routing and communication so that the switches have no processing power at all and function solely as crossbars; as such, switches are only directly connected to servers whereas servers can be connected to both switches and servers. These datacenter networks are called *server-centric* networks with well-known examples being DCell [20], BCube [19], CamCube [1], FiConn [31], DPillar [33], and HCN and BCN [21] (CamCube is somewhat different from the others in that it includes no switches, with servers being directly connected to one another). One criticism of some server-centric datacenter networks is that as they scale, more and more NIC (network interface controller) ports are required at each server; consequently, commodity servers, where there are commonly only two NIC ports, cannot be used and this significantly increases the overall cost. The datacenter network FiConn was explicitly designed to be *dual-port* so that each server is adjacent to exactly one switch and at most one server. From the list of server-centric datacenter networks above, DPillar, HCN, and BCN are also dual-port with other dual-port server-centric datacenter networks having very recently been proposed such as DCube in [22] and SWCube and SWKautz in [32]. It is with dual-port server-centric datacenter networks that we are primarily concerned in this paper.

The scale and expense of datacenters means that the design process needs to be subject to thorough evaluation. As such, a range of metrics has been put forward, again inspired by experiences of interconnection networks in other domains. These metrics attempt to measure network aspects such as hardware costs, scalability, throughput, latency, routing capability, fault tolerance, and so on. However, the characteristics of datacenter networks differ from other interconnection networks and an active research area is the development of appropriate design metrics specifically tailored to datacenters. Fundamental to the evaluation of (datacenter) network designs is their abstraction using mathematics and, in particular, graph theory. Almost all metrics involve an abstraction of a network as a graph, as well as graph-theoretic concepts and techniques. However, as datacenter network design paradigms become more involved, the existing mathematical body of knowledge can fall short. The study of server-centric datacenter networks is a case in point where the abstractions of such networks as graphs are such that there are two very different types of nodes: server-nodes, corresponding to the servers; and switch-nodes, corresponding to the switches. This can complicate the analysis significantly; for example, the exact server-node-to-server-node diameter of DCell is currently not known (see [28] for upper and lower bounds). On the one hand, it is unsatisfactory that basic facts as regards standard server-centric datacenter networks are as yet to be established; but on the other hand, the design of new datacenter networks is giving rise to new and interesting combinatorial questions. It is with such combinatorial questions that we are primarily concerned in this paper.

In [15], a general graph-theoretic construction to build dual-port server-centric datacenter networks was devised. This construction takes an arbitrary graph G as a base graph and builds the *stellar* datacenter network G^* by replacing all vertices of G with switch-nodes and subdividing each edge of G by using two server-nodes. The beauty of this construction is that not only is it widely applicable and easy to describe in a constructive sense, but metric-relevant properties of G^* can be derived in terms of those of G. Consequently, if we carefully choose our base graph G (so that its graph-theoretic properties are potentially well-suited to its use within an interconnection network context) then we can use existing mathematical results concerning G along with the mathematics of the stellar construction in order to derive metric-relevant properties of G^* (as a datacenter network). In [15], such an approach was recently taken with G chosen to be the (well-known and well-studied) generalized hypercube $\mathrm{GQ}_{k,n}$, with the resulting stellar datacenter network, denoted $\mathrm{GQ}^*_{k,n}$, empirically compared against FiConn and DPillar as regards metrics related to network throughput, latency, load balancing, fault-tolerance, and cost-to-build, and with regard to all-to-all, many all-to-all, butterfly, uniform random, hot-region, and hot-spot traffic patterns (generalized hypercubes had already been used to design the datacenter network SWCube in [32]). The stellar datacenter network $\mathrm{GQ}^*_{k,n}$ was shown to generally outperform both FiConn and DPillar (sometimes significantly so), consequently validating the stellar paradigm.

In this paper we continue the study of general stellar datacenter networks but within the context of bisection width. Bisection width is a well-established and generally well-accepted theoretical metric measuring an interconnection network's throughput capacity (it has other relevance too); however, it has recently come under criticism. The extensive, thorough, and novel critique in [25] is of cut-based metrics in general but within a wide network context. Our first contribution in this paper is an examination of the arguments against bisection width in [25] but in a narrow (dual-port) server-centric datacenter network context. We argue that in this narrower context, the arguments of [25] do not carry so much weight and that bisection width, or more precisely the refinement *S-bisection width*, which we detail and justify here, and which is specifically tailored for server-centric datacenter networks, is a relevant throughput metric. Our second contribution is to examine the S-bisection width of a stellar datacenter network G^* in relation to the bisection width of the base graph G. We show that in general these measures can differ but that for a regular graph G, the bisection width of G (almost) always provides an upper bound on the S-bisection width of G^*. Further, we show that this upper bound can be met with

a hypercube as the base graph G; that is, the bisection width of G is equal to the S-bisection width of G^*. Just as the computation of the bisection width of an arbitrary graph G can be difficult and involved, we show that the same is true of the computation of the S-bisection width of G^*. However, we develop a method to compute the S-bisection width of G^* exactly when G is *any* regular graph (of course, regular graphs feature widely as interconnection networks). Our approach to the analysis of S-bisection width is novel in that we exhibit a strong involvement with the well-established mathematical study of *edge isoperimetric problems* in graphs. Our method to compute the S-bisection width of G^* involves the implicit calculation of edge isoperimetric subsets of G. Our third contribution is to apply our methodology and undertake an experimental comparison of $\mathrm{GQ}^*_{k,n}$ and FiConn with respect to S-bisection width. Just as $\mathrm{GQ}^*_{k,n}$ was empirically shown in [15] to have better properties in relation to network throughput, latency, load balancing, fault-tolerance, and cost-to-build than FiConn, we show that the same can be said as regards S-bisection width. An additional generic contribution of this paper is that it emphasises that there is very interesting and relevant mathematics underlying the design of modern datacenter networks.

In Section 2, we outline basic graph-theoretic definitions and concepts as well as detailing the stellar construction and the notion of S-bisection width. In Section 3, we look at bisection width in detail, in tandem with the critique of bisection width in [25], before commenting on this critique and justifying S-bisection width as a valuable throughput metric within our context. Our main technical results are proven in Section 4, with our experimental evaluation of $\mathrm{GQ}^*_{k,n}$ and FiConn with respect to S-bisection width contained in Section 5. We give our conclusions and some directions for further research in Section 6.

2. Basic notation and concepts

We begin by explaining our basic notation and some core graph-theoretic concepts for what follows (other definitions and concepts are introduced as appropriate). Our graphs are always undirected and a graph $G = (V, E)$ has the set V as its vertex-set and the set E as its edge-set. We refer the reader to [12] for notions as regards graph theory and to [10] for background as regards general interconnection networks. We refer to a graph as having vertices and edges, and to an interconnection network (and a datacenter network) as having nodes and links so as to accentuate the fact that it is to act as a communication fabric, with links regarded as full duplex and consisting of two oppositely-oriented channels. Throughout, we write 'DCN' to mean 'datacenter network'. Note that DCNs are usually parameterized families of networks; we refer to both the family and family members as a DCN.

2.1. Stellar DCNs

As we have already mentioned, stellar DCNs arose in [15] as a generic construction to formalise the process by which certain server-centric DCNs, such as SWCube, SWKautz, and DPillar, are built, through the use of 'link subdivision' and so that commodity off-the-shelf (COTS) hardware might be utilised (that is, dual-port servers).

Definition 1. Given the graph $G = (V, E)$, the *stellar DCN* $G^* = (W \cup S, E^*)$ is such that its node-set is partitioned into two non-empty disjoint sets: the set of *switch-nodes* W, of which there are $|V|$ in number; and the set of server-nodes S, of which there are $2|E|$ in number. The set of links E^* of G^* is obtained from E: by identifying the switch-nodes of W with the vertices of V; and by replacing each edge of E with a unique path of 3 links so that the start and end nodes are the corresponding switch-nodes and the two interim nodes are unique server-nodes.

Consequently, a stellar DCN G^* has $|V| + 2|E|$ nodes and $3|E|$ links. Moreover: every server-node is adjacent to exactly one switch-node and exactly one server-node; and every switch-node is adjacent to the same number of server-nodes in G^* as the corresponding vertex in G has neighbours (consequently, it is possible to develop stellar DCNs for which the server-nodes possess aspects of symmetry). In short, G^* is obtained by regarding G as a switch-node network and subdividing the edges of G with pairs of server-nodes. It is not difficult to see that every dual-port server-centric DCN for which: there are switch-nodes and server-nodes; every switch-node is adjacent only to server-nodes; and every server-node is adjacent to exactly one switch-node and exactly one server-node, arises from a stellar construction applied to some graph G.

We illustrate a stellar construction in Fig. 1 using the Petersen graph as our graph G. The switch-nodes in G^* are depicted as squares and the server-nodes as circles (the colouring of nodes is an illustration of an S-bisection in the upcoming Definition 4).

2.2. Bisection width

Interconnection network designs are often evaluated (either analytically or empirically) with respect to some form of workload. The most common workload is a *traffic pattern* defined by a communication matrix in which the (i, j)th element denotes the amount of data a source, i, needs to send to a destination, j; each non-zero element is typically called a *flow*. In such traffic patterns, messages are generated without considering any temporal or causal relationship between them. In this paper, we refer to the uniform random and the all-to-all traffic patterns, both of which deal with unit flows; that is, all

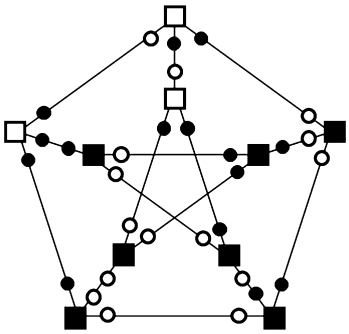

Fig. 1. A stellar construction with a Petersen graph as base. Nodes are black or white to illustrate an S-bisection.

communication matrix entries are 0 or 1. The *uniform random* traffic pattern is obtained by each node of the interconnection network uniformly at random choosing some node (which may be itself) to which to send data. The *all-to-all* traffic pattern is obtained by every node having to send (possibly different) data to every other node. Henceforth, we assume that any traffic pattern only involves unit flows, unless we state otherwise.

As we shall see in the next section, the bisection width of a graph is strongly related with the throughput of an interconnection network under the uniform random and all-to-all traffic patterns.

Definition 2. Let $G = (V, E)$ be some graph. A *cut* in G is a set of edges resulting from a partition (R, T) of V into two non-empty disjoint subsets R and T in that the cut, denoted $[R, T]$, consists of those edges with one incident vertex in R and one in T. The *width* of a cut is the number of edges in the cut. A *bisection* is a cut where the size of R and T differ by at most 1. The *bisection width* $bw(G)$ of G is the smallest width of any bisection.

The concepts from Definition 2 are naturally inherited by interconnection networks when they are abstracted as graphs, although rather than the width of a cut being of primary concern, within an interconnection network it is usually the bandwidth.

Definition 3. Let $G = (V, E)$ be an interconnection network where the two channels corresponding to every link have an associated bandwidth. The *bandwidth* of a cut is the sum of the bandwidths of the channels of the cut, with the *bisection bandwidth* of G being the smallest bandwidth of any bisection.

In what follows, we assume that the channel bandwidth is constant throughout the whole network and we concentrate on the number of links (as opposed to channels) in a cut; that is, we focus on the bisection width of the undirected graph underlying an interconnection network. Also, note that it might be the bisection width, rather than the cumulative bandwidth, that is actually the resource of interest in an interconnection network; for example, as regards the diversity for route selection (across a bisection) or as regards connectivity in the presence of faults. In any case, if all channels of an interconnection network have the same bandwidth b then the bisection bandwidth is simply $2b$ times the bisection width (as there are two channels per link).

2.3. S-bisection width

However, whilst bisection width and bisection bandwidth are relevant to interconnection networks where nodes are homogeneous (in that every node can send and receive messages), these concepts need to be refined for interconnection networks consisting of both server-nodes and switch-nodes (where only server-nodes can send and receive messages, with the switch-nodes providing only message transit). Of course, this is the case for stellar DCNs.

Definition 4. Let $G = (V, E)$ be an undirected graph and let $G^* = (W \cup S, E^*)$ be the corresponding stellar DCN. An *S-partition* of G^* is a partition of the nodes of $W \cup S$ so that: W is partitioned as $R_W \cup T_W$; and S is partitioned as $R_S \cup T_S$ with $|R_S| = |T_S|$. An S-partition $(R_W \cup R_S, T_W \cup T_S)$ yields the *S-bisection* $[R_W \cup R_S, T_W \cup T_S]$ defined as the set of links incident with one node in $R_W \cup R_S$ and one node in $T_W \cup T_S$. The *width* of an S-bisection is the number of links it contains. The *S-bisection width* $bw_S(G^*)$ of G^* is the minimum width over all S-bisections of G^*.

An S-bisection is illustrated in Fig. 1 where server-nodes and switch-nodes are coloured black or white depending upon whether they are in $R_W \cup R_S$ or $T_W \cup T_S$, respectively. Note that necessarily $|R_S| = |T_S| = 15$, although $|R_W| \neq |T_W|$.

Of course, the notion of S-bisection width is applicable to any datacenter network consisting of server-nodes and switch-nodes. We return to a discussion of the S-bisection width of a stellar DCN when we examine the bisection width in relation to interconnection networks in the next section.

2.4. Generalized hypercubes

We shall apply the methodology we develop to generalized hypercubes.

Definition 5. The *generalized hypercube* $\mathrm{GQ}_{k,n}$ of radix n and dimension k is the graph with vertex-set $\{0, 1, \ldots, n-1\}^k$ and where a pair of vertices are adjacent if, and only if, their names differ in exactly one coordinate.

Variations on the generalised hypercube have been proposed in various networking contexts, including [2,8,19,26,33], primarily because generalized hypercubes have good networking capacity, fault tolerance, and bisection width; also, the DCN SWCube [32] was built from generalized hypercubes by subdividing edges not with two server-nodes but with one. Whilst subdividing with one server-node potentially retains symmetry properties of the base graph, subdividing with two, as we do, enables us to not only potentially retain symmetry but also to pack more server-nodes into the resulting datacenter network; we have more to say on this point in our conclusions and directions for further research. In this paper, we instantiate the stellar construction using generalized hypercubes for essentially the same reasons as above and also because stellar generalized hypercubes, $\mathrm{GQ}^*_{k,n}$, have already been studied in a DCN context in [15], as we explained in the Introduction. The family of stellar generalized hypercubes is denoted GQ^*.

3. Bisection bandwidth

In this section we examine the derivation and traditional role of the bisection bandwidth as a metric in the design of interconnection networks before considering a recent critique of the validity of bisection bandwidth as a network metric. Our primary context is when bisection bandwidth is used as a server-centric DCN metric. We then justify our focus on S-bisection width.

3.1. An historical perspective

The bisection width of a graph first found application within the realm of interconnection networks as a means to provide lower bounds on the area occupied by VLSI circuits ([41], where the larger the bisection width of the underlying communication graph, the greater the area required by the corresponding VLSI circuit). The bisection width is also relevant to more general network problems whose algorithmic solutions are structured around the divide-and-conquer paradigm where: the network is split into two halves by the removal of a bisection; the two halves are recursively dealt with; and the bisection is reintroduced. Often the quality of such an algorithm depends upon the size of the bisection, with the smaller the bisection, the better the performance (see, *e.g.*, [9]). A network problem to which the above applies is the network layout problem where a network is typically laid out in a recursive fashion. Whilst this problem was first studied in the context of VLSI layout (see, *e.g.*, [7]), it is relevant to the layout of any network, including DCNs.

However, perhaps the most important application of the bisection width of a graph is with respect to the throughput of an interconnection network under the uniform random and all-to-all traffic patterns (remember: we assume all traffic patterns involve unit flows). As is explained in [10], given some interconnection network and some traffic pattern, the *channel load* γ_c of some channel c (of some link) is the number of flows using that channel, when the flows have been routed (according to some routing algorithm). A *bottleneck channel* is a most heavily loaded channel and we denote its load by γ_{max}. If each channel has bandwidth b then network throughput, namely the amount of data that can be injected at each node without causing a channel to become saturated, is at most $\frac{b}{\gamma_{max}}$. This is the *ideal throughput* of the network. Consider the uniform random traffic pattern and some bisection of smallest width, consisting of β links, in an interconnection network on N nodes. On average, half the flows from one side of the cut use cut-channels to reach the other side of the cut; hence, on average, at least one channel of the bisection is involved in at least $\frac{N}{4\beta}$ flows; that is, $\frac{b}{\gamma_{max}} \leq \frac{4\beta b}{N}$. Consequently, the higher the bisection width in relation to the number of nodes, the better the ideal throughput (theoretically speaking, that is, in that it is implicitly assumed that flows can be balanced across the channels in a bisection, no account is taken of the sizes of flows, no particular routing algorithm is assumed, and the analysis is according to expected performance). Thus, we obtain an upper bound on the expected ideal throughput under the uniform random traffic pattern. An analogous argument can be made for the all-to-all traffic pattern so that, likewise, we obtain an upper bound on the ideal throughput. Note how whilst we strive for a high bisection width so as to maximise (this theoretical) throughput, this is at odds with the consideration of bisection width to minimize VLSI circuit layout or improve the performance of divide-and-conquer network algorithms.

Irrespective of the relevance of the bisection width of a graph to the design of interconnection networks, the **NP**-hardness of computing the bisection width of an arbitrary graph (see, *e.g.*, [17]) has resulted in a thriving area of research as regards deriving the bisection width of (specific classes of) graphs and developing approximation algorithms for the computation of the bisection width in general (see, *e.g.*, [11]). The study of the bisection width is but one aspect of the more general study of isoperimetric problems (about which we say more later) and partitioning problems (see, *e.g.*, [6,35]).

3.2. A recent critique

Irrespective of any difficulties hinted at in the preceding paragraph, it cannot be denied that the bisection width is an established metric as regards the assessment of (datacenter) network designs with respect to throughput. However, there has been a recent thorough and extensive critique in [25] of the role of bisection width and other cut-based metrics as useful metrics for throughput assessment in (general) network design.

The essential content of [25] is as follows. The framework of the paper is wide-ranging, encompassing: the server-centric DCNs DCell [20] and BCube [19] (note that no dual-port server-centric DCNs are considered); the switch-centric DCNs Fat-Tree [3], HyperX [2], Jellyfish [38], Long Hop [42], and Slim Fly [5]; the indirect interconnection networks Dragonfly [27] and Flattened Butterfly [26]; and the direct interconnection network consisting of the family of hypercubes. A 'longest matching' traffic matrix is devised which is used to approximate near worst-case traffic ('worst-case' with respect to throughput). Intuitively speaking, the traffic matrix is constructed so as to 'force' the use of long paths, under the intuition that long paths decrease throughput. This traffic matrix does not describe unit flows, as is the case for us, but results in traffic patterns (including the uniform random and all-to-all traffic pattern) with variably weighted flows. For particular instances of the above networks (with up to around 3,000 servers), the throughput under the new traffic patterns is empirically compared with that predicted using cut-based metrics such as bisection bandwidth. It is argued that the bisection bandwidth and other cut-based metrics generally do not always reflect the worst-case throughput. Other additional limitations of cut-based metrics are remarked upon, such as: they are tied to the uniform random and all-to-all traffic patterns; they only provide 'loose' bounds; and it is **NP**-hard to compute the bisection width of an arbitrary graph. Finally, the above networks are evaluated according to the new approach to evaluating worst-case throughput.

The paper [25] is extremely interesting and an excellent start in attempting to assess worst-case throughput in a systematic fashion. However, in our view, its findings are not conclusive in the much more restricted world in which we operate in this paper, namely with regard to (dual-port) server-centric DCNs. First, no dual-port server-centric DCNs are considered in [25]; indeed, only two server-centric DCNs appear there, whereas many more appear in the literature. Further, the particular instances of server-centric DCNs from DCell and BCube that were considered in [25] contain no more than around 2,000 servers; in our paper, we are concerned with using the server-centric paradigm to ultimately build datacenters of perhaps 1 million servers. As regards the first two of the additional limitations of cut-based metrics that were remarked upon in [25], both are valid observations. However, we have some comments in mitigation. First, although the bisection width is tied to evaluating throughput under the uniform random and all-to-all traffic patterns, all-to-all traffic patterns are extremely important for datacenters within, for example, the MapReduce paradigm. Second, any structural metric, like bisection width, will almost by definition be 'loose', given its ignorance of network applications, flows and their sizes, routing algorithms, load balancing, and so on; although, the structured and symmetric nature of server-centric DCNs perhaps lessens this 'looseness'. Finally, the third of the additional limitations (above) is perhaps a red herring, for the **NP**-hardness result presupposes arbitrary input graphs, whereas server-centric DCNs are all highly structured. There are no known **NP**-hardness results restricted to classes of structured graphs; though it should be noted that it is not always easy to determine the exact bisection width of even simple families of interconnection networks (see, *e.g.*, [4]). This difficulty is accentuated when working with more complicated cut-based metrics, and we are taken to the mathematically involved world of isoperimetric and partitioning problems. A point in mitigation is that often (good) upper and lower bounds are readily available and also heuristic methods can be applied so as to yield reasonable approximations.

In summary, we feel that the objections to bisection width made in [25] have some general validity but that in our much more restricted world of dual-port server-centric DCNs, this validity has yet to be fully established. Consequently, we feel that the study of bisection width as a throughput metric for (dual-port) server-centric DCNs is appropriate. In any case, the bisection width still yields an upper bound on the subsequent throughput under an all-to-all traffic pattern (though arguably there might exist improved upper bounds). Nevertheless, we feel that [25] has made a significant step forward in proposing an alternative methodology for assessing worst-case throughput in DCNs and that this should be built on in future. As a final remark, and importantly, the research in this paper is primarily combinatorial and is inspired (as much of theory is) by applications rather than having as an intention an immediate improvement to applications.

3.3. S-bisection width and its validity

Having discussed and justified bisection width as an interconnection network design metric (in the right circumstances), let us now turn to how we interpret bisection width in DCNs where there are both server-nodes and switch-nodes using S-bisection width. The concept of interconnection networks consisting of switches and processors is, of course, not new. In [10], Dally and Towles cope with such networks by insisting that bisections must partition equally both the set of switches and the set of processors (or terminals in their language). However, to our knowledge, this 'dual-bisectioning' has not been justified and in practice researchers often insist only on a bisection of the terminals; that is, they adopt our S-bisection approach. Indeed, in relation to throughput analysis, if one were to replace the arguments made in this section in favour of bisection width as a valid metric for structured and symmetric interconnection networks with analogous ones for server-centric DCNs but with respect to S-bisection width then one would obtain a valid justification for S-bisection width as a sensible metric in our server-centric DCNs. This justification is further strengthened by the prevalence of all-to-all traffic patterns built around the MapReduce paradigm in datacenter usage.

There is an alternative as to what should constitute the correct notion of a bisection in a server-centric DCN. In [4], and with respect to the server-centric DCN BCube, two models are adopted: one where a switch-node is joined directly to its adjacent server-nodes (as is the case for us); and one where a switch-node is replaced by what is called a hyperlink. It is argued that the first model is appropriate for the situation where the links provide bottlenecks and the second where the switches provide bottlenecks. Whilst the second model is theoretically interesting, we do not discuss it further here, as most modern networking equipment is non-oversubscribed, *e.g.*, 32Gbps 24-port switches for 1Gbps Ethernet and 480Gbps 32-port switches for 10Gbps Ethernet.

4. Bisection width vs. S-bisection width

We are now in a position whereby we have the validated server-centric DCN metric S-bisection width and we wish to evaluate the S-bisection width of a stellar DCN G^*, namely $bw_S(G^*)$, by relating it to the bisection width of the base graph G, namely $bw(G)$. We first give some general results before fixing G to be a hypercube and obtaining precise results for stellar hypercubes. We then develop a general technique to compute the S-bisection width of an arbitrary regular graph and apply this general technique to generalized hypercubes. Throughout, we see a close relationship between the S-bisection width of stellar DCNs and edge isoperimetric properties.

4.1. Some basic results

The following two basic results are proven in [15]. Note that in a server-centric DCN the length (also known as the hop-length) of a path from one server-node to another server-node is usually taken to be the number of server-nodes on this path (this reflects the fact that switches generally operate as crossbars and merely forward on packets with negligible buffering or routing overheads, when compared with these imposed by servers).

Lemma 6 *([15]). Let $G = (V, E)$ be a connected graph with u' and v' distinct vertices of V so that a shortest path in G from u' to v' has length m. Also, let u and v be server-nodes of G^* so that u and v are adjacent to the switch-nodes u' and v', respectively. The length of a shortest path from u to v in G^* is $2m-1$, $2m$, or $2m+1$. Hence, if G has diameter δ then G^* has diameter at most $2\delta+1$.*

The connectivity of a graph is equal to the minimum number of mutually internally vertex-disjoint paths between any 2 distinct vertices. Clearly, at most two internally node-disjoint paths exist between a pair of server-nodes in G^*, since server-nodes have degree 2, but the connectivity of G is retained for switch-nodes within G^*.

Lemma 7 *([15]). Let G be a graph of connectivity $c \geq 1$. Let u and v be distinct server-nodes of G^* that are not adjacent to the same switch-node. There are c mutually server-node-disjoint paths in G^* from u to v so that no switch-node apart from the two switch-nodes adjacent to u and v lies on more than one of these paths.*

The stellar DCN G^* inherits any routing algorithm that exists for G since a path in G can be subdivided to induce a corresponding switch-node-to-switch-node path in G^*. A path between two server-nodes in G^* can therefore be found by, roughly speaking, augmenting a path between their respective neighbouring switch-nodes.

While notions of diameter, connectivity, and routing are straightforward to derive for G^*, directly from G, the situation is different in relation to relating the bisection width of G and the S-bisection width of G^*. Exploring this relationship is what this paper is all about.

Interconnection networks can, more often than not, be abstracted as regular graphs and there is a fundamental relationship between the bisection width of a regular graph G and the S-bisection width of the stellar DCN G^*.

Lemma 8. *Let $G = (V, E)$ be a d-regular graph resulting in the stellar DCN G^*. If $|V|$ is even or $bw(G) \geq \frac{d}{2}$ then $bw_S(G^*) \leq bw(G)$. If $|V|$ is odd and $bw(G) < \frac{d}{2}$ then*

$$bw_S(G^*) \leq \begin{cases} \frac{d}{2} & \text{if } \frac{d}{2} - bw(G) \text{ is even} \\ \frac{d}{2}+1 & \text{if } \frac{d}{2} - bw(G) \text{ is odd.} \end{cases}$$

Proof. Suppose that $G^* = (S \cup W, E)$. Let (R, T) be a bisection of G. Build a partition (R', T') of G^* as follows: if $v \in V \cap R$ (resp. $v \in V \cap T$) then place the corresponding switch-node v and all its d adjacent server-nodes in R' (resp. T').

Case (a): $|V|$ is even.

So $|R' \cap S| = |T' \cap S| = \frac{d|V|}{2}$ and $|[R', T']| = |[R, T]|$. Hence, $bw_S(G^*) \leq bw(G)$.

Case (b): $|V|$ is odd.

As G is regular, d must be even, and w.l.o.g. $|R' \cap S| = \frac{d(|V|+1)}{2}$ and $|T' \cap S| = \frac{d(|V|-1)}{2}$. Choose $\min\{bw(G), \frac{d}{2}\}$ pairs of adjacent server-nodes so that one of these server-nodes lies in R' and one lies in T', and for each pair, move the server-node that was in R' to T'.

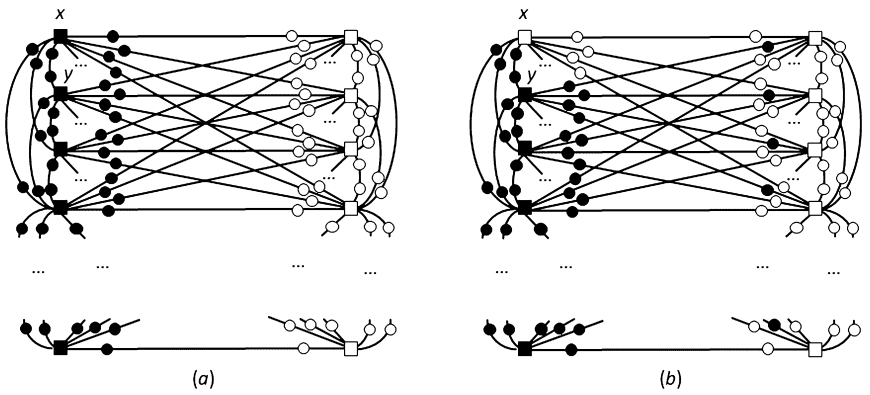

Fig. 2. Amending a bisection in K^*_{2n}.

If $bw(G) \geq \frac{d}{2}$ then we now have that $|R' \cap S| = |T' \cap S| = \frac{d|V|}{2}$ and $|[R', T']| = |[R, T]|$; consequently, $bw_S(G^*) \leq bw(G)$.

If $bw(G) < \frac{d}{2}$ then we have that $|R' \cap S| = \frac{d(|V|+1)}{2} - bw(G)$ and $|T' \cap S| = \frac{d(|V|-1)}{2} + bw(G)$, with $|[R', T']| = |[R, T]|$. Choose an additional $\lfloor \frac{\frac{d}{2} - bw(G)}{2} \rfloor$ pairs of adjacent server-nodes in R' and move them to T'. When $\frac{d}{2} - bw(G)$ is even, we have that $|R' \cap S| = |T' \cap S| = \frac{d|V|}{2}$ and $|[R', T']| = |[R, T]| + \frac{d}{2} - bw(G)$; consequently, $bw_S(G^*) \leq \frac{d}{2}$. When $\frac{d}{2} - bw(G)$ is odd, we have that $|R' \cap S| = |T' \cap S| + 2$ and $|[R', T']| = |[R, T]| + \frac{d}{2} - bw(G) - 1$; so, we need to choose an additional server-node of R' and move it to T' which yields that $|R' \cap S| = |T' \cap S| = \frac{d|V|}{2}$ and $|[R', T']| = |[R, T]| + \frac{d}{2} - bw(G) + 1$, with $bw_S(G^*) \leq \frac{d}{2} + 1$. □

Of course, regular interconnection networks invariably have a bisection width that is at least half their degree (indeed, the degree of an interconnection network is usually such as to make the bisection width relatively large) and so the simple upper bound for $bw_S(G^*)$ of $bw(G)$ from Lemma 8 applies. If $G = (V, E)$ is regular of degree d so that $bw(G) < \frac{d}{2}$ then we say that the bisection width of G is *small*. Given the scenario where our aim is to maximise the S-bisection width of G^*, Lemma 8 yields that when G is regular and either $|V|$ is even or $bw(G)$ is not small, the best we can hope to do is to show that $bw_S(G^*) = bw(G)$. However, this need not be the case, even for the complete graph.

Lemma 9. *For all $n \geq 2$, it is the case that $bw_S(K^*_{2n}) \leq bw(K_{2n}) - \lfloor \frac{n}{2} \rfloor$.*

Proof. Suppose that $K^*_{2n} = (S \cup W, E^*)$. Take some bisection (R, T) of K_{2n} of (minimal) width n^2 and build the S-bisection (R', T') of K^*_{2n} as we did in the proof of Lemma 8. Let $x, y \in R' \cap W$ be such that $x \neq y$. Note that x is adjacent to: n server-nodes of R', denoted by the set X_R, each of which is adjacent to a server-node in T'; and $n-1$ server-nodes of R', each of which is adjacent to a server-node in R'. Note also that y is adjacent to: n server-nodes of R', each of which is adjacent to a server-node in T', with this resulting set of server-nodes of T' denoted Y_T; and $n-1$ server-nodes of R', each of which is adjacent to a server-node in R'. The situation can be visualized as in Fig. 2(*a*). Switch-nodes are depicted by squares and server-nodes by circles. The S-bisection (R', T') is depicted so that nodes of R' are black and the nodes of T' are white.

Amend the S-bisection (R', T') of K^*_{2n} as follows: move the switch-node x and the server-nodes of X_R to T'; and move the server-nodes of Y_T to R'. Fig. 2(*b*) depicts this amendment where: some black nodes (the switch-node x and the n server-nodes of X_R) have changed colour to white; and some white nodes (the n server-nodes of Y_T) have changed colour to black. We obtain a new S-bisection (as there are still exactly the same number of server-nodes in the two new subsets R' and T'; that is, exactly the same number of server-nodes coloured black and white) and this new S-bisection has width $n^2 - 1$. By pairing the (original) switch-nodes of R' in $\lfloor \frac{n}{2} \rfloor$ pairs and iterating the above amendment, we repeatedly obtain an S-bisection of K^*_{2n} of width one less than the previous one. After $\lfloor \frac{n}{2} \rfloor$ iterations we have exhausted all pairs of switch-nodes and we have an S-bisection of width $n^2 - \lfloor \frac{n}{2} \rfloor$. □

4.2. The stellar hypercube

We now turn our attention to stellar hypercubes, the reason being that hypercubes are the most ubiquitous interconnection networks and generally form the starting point of any investigation. We prove that the bisection width of the hypercube Q_n and the S-bisection width of Q^*_n are the same; so, for a standard class of interconnection networks the S-bisection width of the stellar DCN can meet the upper bound set in Lemma 8 by the bisection width of the base graph.

In order that we might do this, we need to provide some notions and results relating to edge isoperimetric problems in graphs.

Let $G = (V, E)$ be a connected graph. For a subset $A \subseteq V$, define the sets of edges $I_G(A) = \{(u, v) \in E : u, v \in A\}$ and $\theta_G(A) = \{(u, v) \in E : u \in A, v \notin A\}$. For any k where $1 \leq k \leq |V|$, define $I_G(k)$ as $\max\{|I_G(A)| : A \subseteq V, |A| = k\}$ and $\theta_G(k)$ as $\min\{|\theta_G(A)| : A \subseteq V, |A| = k\}$. Note that if G is regular of degree d then $2I_G(k) + \theta_G(k) = kd$. A subset $A \subseteq V$ where either $|I_G(A)| = I_G(k)$ or $|\theta_G(A)| = \theta_G(k)$ is called an *edge isoperimetric subset* of V. The study of (edge) isoperimetric sets in combinatorial mathematics has a long history and includes ascertaining the functions $I_G(k)$ and $\theta_G(k)$ for specific graphs G (isoperimetric problems are surveyed in, *e.g.*, [6]).

We also require the following definitions. Fix n. For an integer i where $0 \leq i \leq 2^n - 1$, we denote the number of 1's in the representation of i as an n-bit binary string by $wt(i)$. The *lexicographic order* of elements of $\{0, 1\}^n$ is such that the n-bit binary string b is less than the n-bit binary string b' if there is an index $j \in \{1, 2, \ldots, n\}$ such that the first $j - 1$ bits of b and b' are identical, the jth bit of b is 0 and the jth bit of b' is 1.

We use the following result, first established by Harper in 1964.

Theorem 10 *([24]). Let $n \geq 1$ and $1 \leq k \leq 2^n$. A particular subset of vertices $A \subseteq V(Q_n)$ of size k that minimizes $|I_{Q_n}(A)|$ is the initial segment of size k of the lexicographically-ordered elements of $\{0, 1\}^n$, and*

$$I_{Q_n}(k) = \sum_{i=0}^{k-1} wt(i).$$

As a matter of fact, a great many results relating to edge isoperimetric problems in graphs involve initial segments of 'lexicographically-ordered' vertices (where the notion of 'lexicographic-order' varies from graph to graph; see, *e.g.*, [6]).

We also use a particular property of the function $I_{Q_n}(k)$.

Lemma 11. *Fix n. Let $k = 2^m + x$ where $0 \leq m < n$ and $0 \leq x \leq 2^m$. We have that $I_{Q_n}(k) = I_{Q_n}(2^m) + x + I_{Q_n}(x)$. Hence, $I_{Q_n}(2^m) = m2^{m-1}$.*

Proof. By Theorem 10, $I_{Q_n}(k) = \sum_{i=0}^{k-1} wt(i)$. The first 2^m elements of the lexicographically-ordered elements of $\{0, 1\}^*$ are

$$00\ldots0\underline{0}\ldots\underline{000}, \quad 00\ldots0\underline{0}\ldots\underline{001}, \quad 00\ldots0\underline{0}\ldots\underline{010}, \quad 00\ldots0\underline{0}\ldots\underline{011},$$
$$00\ldots0\underline{0}\ldots\underline{100}, \quad \ldots \quad 00\ldots0\underline{1}\ldots\underline{111},$$

where the last m bits are underlined, and the next x elements are

$$00\ldots1\underline{0}\ldots\underline{000}, \quad 00\ldots1\underline{0}\ldots\underline{001}, \quad 00\ldots1\underline{0}\ldots\underline{010}, \quad 00\ldots1\underline{0}\ldots\underline{011},$$
$$00\ldots1\underline{0}\ldots\underline{100}, \quad \ldots$$

Consequently, counting the number of 1's in the strings of the first block yields $I_{Q_n}(2^m)$ whereas counting the number of 1's in the strings of the second block yields $x + I_{Q_n}(x)$. Hence, $I_{Q_n}(k) = I_{Q_n}(2^m) + x + I_{Q_n}(x)$. The fact that $I_{Q_n}(2^m) = m2^{m-1}$ follows from a simple induction. □

Now for the first of our main results.

Theorem 12. *The S-bisection width of Q_n^* is equal to 2^{n-1}, which is also the bisection width of Q_n.*

Proof. The bisection width of Q_n is well known to be 2^{n-1} (see, *e.g.*, [29]). Suppose that $Q_n^* = (S \cup W, E^*)$. Let φ be the natural isomorphism from the switch-nodes W of Q_n^* to the vertices of Q_n; that is, if u is a switch-node (resp. U is a set of switch-nodes) of W then $\varphi(u)$ (resp. $\varphi(U)$) is the corresponding vertex (resp. set of vertices) of Q_n, and if u' is a vertex (resp. U' is a set of vertices) of Q_n then $\varphi^{-1}(u')$ (resp. $\varphi^{-1}(U')$) is the corresponding switch-node (resp. set of switch-nodes) of W.

The result is trivial for $n = 1$ (partition the nodes of Q_1^* so that a server-node and its adjacent switch-node are on one side of the partition, with the other server-node and switch-node on the other). So we may assume throughout that $n \geq 2$.

Given a partition (R, T) of either Q_n or Q_n^*, we find it helpful to think of this partition as a 2-colouring of the vertices or nodes, according to whether the vertex or node lies in R or T. Conversely, any 2-colouring of Q_n or Q_n^* corresponds to a partition.

Let (R, T) be a partition of the vertices of Q_n so that we include in R all those vertices whose first component (in their n-bit names) is 0 and in T all those vertices whose first component is 1. In Q_n^*, colour all switch-nodes of $\varphi^{-1}(R)$ white and all switch-nodes of $\varphi^{-1}(T)$ black. Extend this colouring by colouring every server-node with the colour of its (unique)

adjacent switch-node. This results in an S-bisection of width 2^{n-1}. In the remainder of this proof, we prove that there cannot exist an S-bisection of Q_n^* of width less than 2^{n-1}.

Let π be a partition of the server-nodes of Q_n^* so that exactly half are coloured black and half white; denote these sets of server-nodes by S_b and S_w, respectively. Hence, $|S_b| = |S_w| = n2^{n-1}$. The partition π induces a colouring of the switch-nodes of Q_n^*: a switch-node is coloured with the colour colouring the majority of the switch-node's adjacent server-nodes, with ties broken arbitrarily. Let the set of switch-nodes coloured black be denoted W_b with the set of switch-nodes coloured white denoted W_w. Hence, $|W_b| + |W_w| = 2^n$ and we have an S-bisection of Q_n^* which we call π also.

We claim that any S-bisection of Q_n^* of minimum width must have been constructed as π was above. Suppose otherwise: either there must exist a switch-node that is coloured black when more than one half of its neighbouring server-nodes are coloured white; or there is a switch-node that is coloured white when more than one half of its neighbouring server-nodes are coloured black. Either way, by swapping the colour of this switch-node we obtain an S-bisection of Q_n^* of smaller width, which yields a contradiction.

As our working hypothesis, suppose that π is an S-bisection of Q_n^* of minimal width (constructed as above) and that this width is less than 2^{n-1}. We will show that this yields a contradiction. In order to do this, we differentiate between two cases: the first where $|W_w| \neq |W_b|$; and the second where $|W_w| = |W_b|$.

Case 1: Suppose w.l.o.g. that $|W_w| < |W_b|$; hence, $|W_w| < 2^{n-1}$ (the case where $|W_b| < |W_w|$ proceeds identically).

The colouring of the switch-nodes of Q_n^* naturally induces a colouring of the vertices of Q_n, with the set of vertices of Q_n that are coloured white (resp. black) being denoted W'_w (resp. W'_b). In particular, $\varphi(W_w) = W'_w$ and $\varphi(W_b) = W'_b$, with $|W_w| = |W'_w|$ and $|W_b| = |W'_b|$.

Consider the subgraph G_w of Q_n induced by the vertices of W'_w. By definition, the number of edges in G_w is at most $I_{Q_n}(|W_w|)$. Let X denote the set of edges of Q_n that are incident with exactly one vertex of W'_w. So, $n|W_w| \leq 2I_{Q_n}(|W_w|) + |X|$. Let $(u, v) \in X$, with $u \in W'_w$ and $v \in W'_b$. The (unique) path of length 3 in Q_n^* joining $\varphi^{-1}(u) \in W_w$ and $\varphi^{-1}(v) \in W_b$ is such that at least one link on this path is incident with nodes of different colours. Hence, every edge of X corresponds to at least one link of Q_n^* whose incident nodes have different colours, and no link of Q_n^* whose incident nodes have different colours stems from two different edges of X. Thus, $|X| < 2^{n-1}$ and consequently $2^{n-1} > n|W_w| - 2I_{Q_n}(|W_w|)$.

We claim that $|W_w| < 2^{n-2}$. Suppose otherwise and that $|W_w| = 2^{n-2} + x$, where $0 \leq x < 2^{n-2}$. By Lemma 11, we have that

$$\begin{aligned} 2^{n-1} &> n|W_w| - 2I_{Q_n}(|W_w|) = n(2^{n-2} + x) - 2I_{Q_n}(2^{n-2} + x) \\ &= n(2^{n-2} + x) - 2(I_{Q_n}(2^{n-2}) + x + I_{Q_n}(x)) = n(2^{n-2} + x) - 2((n-2)2^{n-3} + x + I_{Q_n}(x)) \\ &= 2^{n-1} + x(n-2) - 2I_{Q_n}(x). \end{aligned}$$

Hence, $I_{Q_n}(x) > \frac{x(n-2)}{2}$.

Lemma 13. *Let $m \geq 1$. If $0 < x < 2^m$ then $I_{Q_n}(x) \leq \frac{xm - wt(x)}{2}$, with equality only if $x = 2^m - 1$ (in which case $I_{Q_n}(2^m - 1) = m(2^{m-1} - 1)$).*

Proof. Suppose that $x = 2^{i_1} + 2^{i_2} + \ldots + 2^{i_k}$, for some k where $1 \leq k \leq m$ and where $m - 1 \geq i_1 > i_2 > \ldots > i_k \geq 0$. Consider the matrix with rows indexed 0 to $x - 1$ and with columns indexed 1 to m where the entry on row i and in column j is the jth bit of the m-bit binary number representing the decimal number i. Note that the m-bit binary string b on row i is the ith m-bit binary string b' in the lexicographic order on $\{0, 1\}^m$ except written in reverse; in particular, $wt(b) = wt(b')$. Thus, by Theorem 10, $I_{Q_n}(x)$ is the total number of 1s in this matrix. The matrix can be visualized in Fig. 3.

Consider the rows 0 to $2^{i_1} - 1$. On these rows: all 1s appear in columns 1 to $i_1 - 1$ (the upper shaded block in Fig. 3); summing over the columns 1 to $i_1 - 1$, there are exactly the same numbers of 0s and 1s; and in column i_1 there are 2^{i_1} 0s. Note also that in the rows 2^{i_1} to $x - 1$: every entry in columns $i_1 + 1$ to m contains 0; and every entry in column i_1 is a 1 but there are less than 2^{i_1} 1s in this column (the rightmost shaded strip in Fig. 3). Thus, if we ignore the matrix entries on rows 2^{i_1} to $x - 1$ and in columns 1 to $i_1 - 1$ then there are more 0s than there are 1s. Henceforth, we only focus on the entries on rows 2^{i_1} to $x - 1$ and in columns 1 to $i_1 - 1$.

Consider the rows 2^{i_1} to $2^{i_1} + 2^{i_2} - 1$. On these rows: apart from the 1s in column i_1, all 1s appear in columns 1 to $i_2 - 1$ (the middle shaded block in Fig. 3); apart from the 1s in column i_1, summing over columns 1 to $i_2 - 1$, there are exactly the same numbers of 0s and 1s; and in column i_2 there are 2^{i_2} 0s. Note also that in the rows $2^{i_1} + 2^{i_2}$ to $x - 1$: every entry in columns $i_2 + 1$ to $i_1 - 1$ contains 0; and every entry in column i_2 is a 1 but there are less than 2^{i_2} 1s in this column (the middle shaded strip in Fig. 3). Thus, if we ignore the matrix entries on rows $2^{i_1} + 2^{i_2}$ to $x - 1$ and in columns 1 to $i_2 - 1$ then there are more 0s than there are 1s.

We can proceed in this fashion with $i_3, i_4, \ldots, i_k$. Let x_0 (resp. x_1) be the number of 0s (resp. 1s) in the matrix. We clearly have that $x_0 + x_1 = xm$ and $x_0 \geq x_1 + wt(x)$ (as there are $wt(x)$ columns where there are more 0s than there are 1s). Thus, $x_1 \leq xm - (x_1 + wt(x))$ and so $I_{Q_n}(x) = x_1 \leq \frac{xm - wt(x)}{2}$. (Note that if $i_k = 0$ then the entry of the matrix on row $x - 1$ and in column 1 is 0.) It is only when $x = 2^m - 1$ that $I_{Q_n}(x) = \frac{xm - wt(x)}{2}$, *i.e.*, $I_{Q_n}(2^m - 1) = m(2^{m-1} - 1)$. The result follows. □

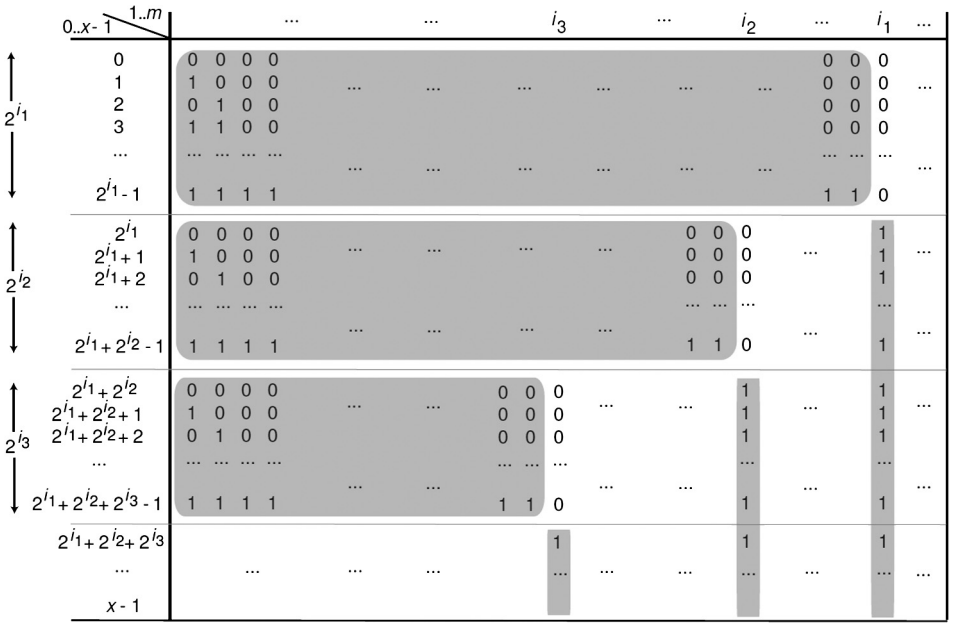

Fig. 3. Matrix of 0s and 1s in the computation of $I_{Q_n}(x)$.

By Lemma 13 and the fact that $x < 2^{n-2}$, we must have that $\frac{x(n-2)}{2} \geq I_{Q_n}(x) > \frac{x(n-2)}{2}$, which yields a contradiction. Hence, our claim follows and we have $|W_w| < 2^{n-2}$.

Consider the set of nodes S_w in Q_n^*. Each node $u \in S_w$ is adjacent to exactly one node $u' \in W$. Call this node u' the node of W that is *tied* to u and the fact that u' is tied to u a *tie*. The number of ties to nodes of W_b is less than 2^{n-1} and there are $n2^{n-1}$ nodes in S_w. Thus, there are more than $n2^{n-1} - 2^{n-1} = (n-1)2^{n-1}$ ties from nodes of S_w to nodes of W_w. But each node of W_w is involved in at most n ties with nodes from S_w and so there must be more than $\frac{(n-1)}{n}2^{n-1}$ nodes in W_w. However, from above, $|W_w| < 2^{n-2}$ which yields a contradiction (as $n > 1$).

Case 2: $|W_w| = |W_b|$; hence, $|W_w| = |W_b| = 2^{n-1}$.

Consider the bisection of Q_n obtained by colouring any vertex $\varphi(u)$ of Q_n with the colour of u in Q_n^*. This bisection has width at least 2^{n-1}. Every edge (u, v) of Q_n where u and v have different colours is such that the unique path of length 3 in Q_n^* joining $\varphi^{-1}(u)$ and $\varphi^{-1}(v)$ is such that at least one of its links is incident with nodes of different colours. Hence, the width of the S-bisection π is at least 2^{n-1}, which yields a contradiction. The result follows. □

4.3. The general case

We now develop a method for computing the S-bisection width of G^* for regular graphs $G = (V, E)$. However, while our method is indeed general, in order to apply this method we need concise information as regards edge isoperimetric subsets of G.

Theorem 14. *Let $G = (V, E)$ be a d-regular graph. The S-bisection width of G^* is given by*

$$\min\{\beta_r : 0 \leq r \leq |V|\},$$

where

$$\beta_r = \begin{cases} rd - 2\lfloor \frac{|E|}{2} \rfloor & \text{if } |E| \leq 2I_G(r) \\ \theta_G(r) & \text{if } 2I_G(r) < |E| < 2I_G(r) + 2\theta_G(r) \\ 2\lceil \frac{|E|}{2} \rceil - rd & \text{if } 2I_G(r) + 2\theta_G(r) \leq |E|. \end{cases}$$

Proof. Fix r so that $0 \leq r \leq |V|$. Consider some subset $R_W \subseteq W$ of r switch-nodes in G^*. Define $T_W = W \setminus R_W$. Every 3-path in G^* joining two switch-nodes is now determined as of one of three types:

(i) both switch-nodes are in R_W
(ii) one switch-node is in R_W and one is in T_W
(iii) both switch-nodes are in T_W.

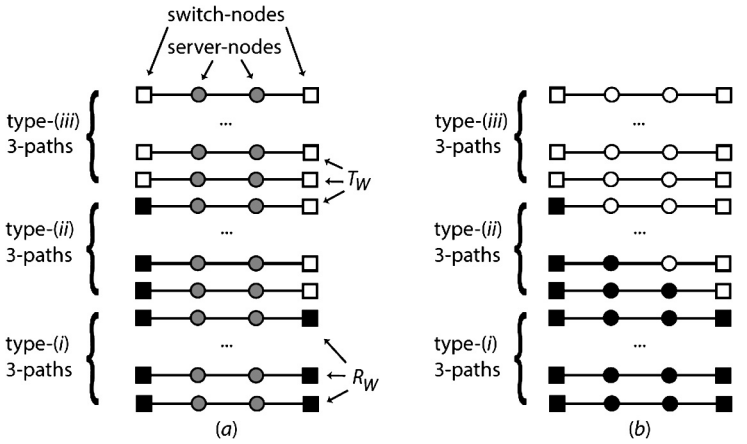

Fig. 4. The set of 3-paths.

We can enumerate the above $|E|$ 3-paths in G^* by ordering all 3-paths of type (i) before all 3-paths of type (ii) before all 3-paths of type (iii), as in Fig. 4(a). Note that in Fig. 4(a) the left and right columns of switch-nodes are such that every switch-node of W appears on d occasions. For convenience, in Fig. 4(a) we draw 3-paths of type (ii) so that the switch-node in R_W appears in the left column, and we depict switch-nodes of R_W as black and switch-nodes of T_W as white, with all server-nodes as grey.

In order to extend the partition (R_W, T_W) of switch-nodes of G^* to an S-bisection $(R_W \cup R_S, T_W \cup T_S)$ of G^*, we have to choose $|E|$ server-nodes of S as R_S, with the remaining $|E|$ server-nodes of S constituting T_S. With reference to Fig. 4(a), this equates to colouring $|E|$ (currently grey) server-nodes black and $|E|$ white. We wish to do this so as to minimize the width of the resulting S-bisection. We make the following observations.

(a) No matter whether we choose to colour the 2 server-nodes of a 3-path of type (ii) both black, black and white, or both white, we can do this so that the number of links so contributed to the total width of the resulting S-bisection is exactly 1 (it is impossible for any such 3-path not to contribute a link to the total width).
(b) Ideally, we would wish to colour both server-nodes of a 3-path of type (i) (resp. (iii)) black (resp. white) so that no links are contributed to the S-bisection. In both cases, if we cannot meet this ideal then no matter how we colour the 2 server-nodes of a 3-path of type (i) or (iii), the number of links so contributed to the total width of the resulting S-bisection is exactly 2.

Consequently, it is apparent that in order to extend the partition (R_W, T_W) to an S-bisection $(R_W \cup R_S, T_W \cup T_S)$ so as to minimize the width, we should:

- first, place as many pairs of server-nodes of 3-paths of type (i) as we can in R_S
- next, place as many pairs of server-nodes of 3-paths of type (iii) as we can in T_S
- finally, place the remaining server-nodes arbitrarily (but so that we ensure any 3-path of type (ii) contributes exactly 1 link to the width of the resulting S-bisection).

We refer to this algorithm as our *colouring algorithm* (we think of a server-node of R_S or T_S as coloured black or white, respectively), with the output as depicted in Fig. 4(b). The width of the resulting S-bisection is determined by the relative numbers of 3-paths of types (i), (ii), and (iii). Moreover, this width can be easily computed.

An obvious algorithm to compute the S-bisection width of G^* springs to mind: for every $r \geq 1$ and for every possible subset $R_W \subseteq W$ with $|R_W| = r$, compute the width of the S-bisection $(R_W \cup R_S, T_W \cup T_S)$ constructed according to our colouring algorithm, and take the minimal such value. Unfortunately, this will not really suffice as the algorithm runs in exponential time. However, we can improve things considerably.

Again, fix $r \geq 1$ and consider the different possibilities for $R_W \subseteq W$ where $|R_W| = r$. Each R_W yields a unique *figurative depiction* (*f-d*) as displayed in Fig. 4(a), whereby rd switch-nodes (in the right and left columns) are coloured black so that in each column the switch-nodes coloured black appear contiguously (starting from the bottom) and so that w.l.o.g. there are at least as many black switch-nodes in the left column as there are in the right. We have some observations.

- Having the f-d corresponding to some subset R_W suffices for us to calculate the minimal width of a bisection of the form $(R_W \cup R_S, T_W \cup T_S)$.
- Different R_W's might yield the same f-d.
- There might exist an f-d that does not stem from any subset $R_W \subseteq W$ of r switch-nodes.

As regards this last observation, this is because the structure of G and the choice of r presents constraints upon a possible f-d that might arise, *e.g.*, if we have an f-d where all black switch-nodes lie in the left column then there must exist an independent set of size r in G. Note that even if an f-d does not stem from some subset $R_W \subseteq W$ of size r, we can still talk about the *width* of the f-d as the number of links incident with nodes of different colours after applying our colouring algorithm to the f-d.

The above suggests an improved algorithm to compute the S-bisection width of G^*: for every $r \geq 1$, generate every f-d stemming from some subset $R_W \subseteq W$ of r switch-nodes, compute the width of these f-d's, and take the minimal such value. Such an algorithm would imply that we need to be able to decide when an f-d stems from a subset $R_W \subseteq W$ of r switch-nodes; however, we can get round this obstacle.

Denote by β_r the minimal width of the S-bisection $(R_W \cup R_S, T_W \cup T_S)$ constructed according to our colouring algorithm, taken over all subsets $R_W \subseteq W$ of r switch-nodes. Order the f-d's, involving rd black switch-nodes and irrespective of whether they stem from some subset R_W, so that some f-d f appears before some f-d f' if, and only if, f has more black switch-nodes in the left column than f' does. For some f-d f, let f_1 (resp. f_2, f_3) denote the number of 3-paths of type (i) (resp. (ii), (iii)). Consequently, we are ordering the f-d's according to increasing f_1. Note that $2f_1 + f_2 = rd$.

Suppose that $|E|$ is even. Let f and f' be consecutive f-d's in our ordering, with f coming before f'; so, $f_2 \geq 2$, $f'_1 = f_1 + 1$, and $f'_2 = f_2 - 2$. Let w (resp. w') be the width of the f-d f (resp. f'). With reference to Fig. 4(b), there are three essential cases.

Case (a): $\frac{|E|}{2} \leq f_1$.
We have $w = 2(f_1 - \frac{|E|}{2}) + f_2 = rd - |E|$ and $w' = w$.

Case (b): $f_1 < \frac{|E|}{2} < f_1 + f_2$.
We have $w = f_2$ and $w' = w - 2$.

Case (c): $f_1 + f_2 \leq \frac{|E|}{2}$.
We have $w = f_2 + 2(\frac{|E|}{2} - (f_1 + f_2)) = |E| - rd$ and $w' = w$.

Consequently, our ordering of the f-d's is such that as we move down the ordering, the width of the f-d's does not increase. In particular, the width of the f-d that stems from some subset $R_W \subseteq W$ of r switch-nodes and is furthest down the ordering is equal to β_r. This f-d is the f-d f for which $f_1 = I_G(r)$, as: f stems from some subset $R_W \subseteq W$ of r switch-nodes; and if any f-d f' for which $f'_1 > f_1$ stems from some subset $R'_W \subseteq W$ of r switch-nodes then $I_G(r) \geq f'_1$, which yields a contradiction. So, if we know $I_G(r)$ then we can trivially obtain β_r as: $rd - |E|$ in Case (a); $rd - 2I_G(r) = \theta_G(r)$ in Case (b); and $|E| - rd$ in Case (c).

Suppose that $|E|$ is odd. Define f, f', w, and w' as above. With reference to Fig. 4(b), there are now five essential cases.

Case (a): $\lceil \frac{|E|}{2} \rceil \leq f_1$.
We have $w = 2(f_1 - \lfloor \frac{|E|}{2} \rfloor) + f_2 = rd - (|E| - 1)$ and $w' = w$.

Case $(b.1)$: $f_1 + 1 = \lceil \frac{|E|}{2} \rceil$.
We have $w = f_2$ and $w' = w$.

Case $(b.2)$: $f_1 + 1 < \lceil \frac{|E|}{2} \rceil < f_1 + f_2$.
We have $w = f_2$ and $w' = w - 2$.

Case $(b.3)$: $\lceil \frac{|E|}{2} \rceil = f_1 + f_2$.
We have $w = f_2$ and $w' = w$.

Case (c): $f_1 + f_2 < \lceil \frac{|E|}{2} \rceil$.
We have $w = f_2 + 2(\lceil \frac{|E|}{2} \rceil - (f_1 + f_2)) = |E| + 1 - rd$ and $w' = w$.

As above, the f-d f for which $f_1 = I_G(r)$ has width β_r. As $2I_G(r) + \theta_G(r) = rd$, the result follows. □

Of course, in order to apply Theorem 14 we need edge isoperimetric information that might not be readily available; however, sometimes it is. Our application of Theorem 14 is to the stellar generalized hypercube $\mathrm{GQ}^*_{k,n}$. Clearly, $\mathrm{GQ}^*_{k,n}$ has $k(n-1)n^k$ server-nodes and n^k switch-nodes with each switch-node having degree $k(n-1)$. In order to apply Theorem 14 we need to know $I_{\mathrm{GQ}_{k,n}}(t)$ (or, equivalently, $\theta_{\mathrm{GQ}_{k,n}}(t)$).

Theorem 15 *([34,36]). For $1 \leq t \leq n^k$, $I_{\mathrm{GQ}_{k,n}}(t) = \sum_{i=0}^{t-1} wt_n(i)$, where $wt_n(i)$ is the sum of the k (base n) 'digits' of i.*

Nakano also shows in [36] that $I_{\mathrm{GQ}_{k,n}}(t)$ evaluates to

$$I_{\mathrm{GQ}_{k,n}}(t) = \begin{cases} \binom{t}{2} & \text{if } t \leq n \\ \sum_{i=0}^{n-1} \left(I_{\mathrm{GQ}_{k,n}}\left(\lfloor \frac{t+i}{n} \rfloor \right) + (n-i-1) \lfloor \frac{t+i}{n} \rfloor \right) & \text{if } t > n. \end{cases}$$

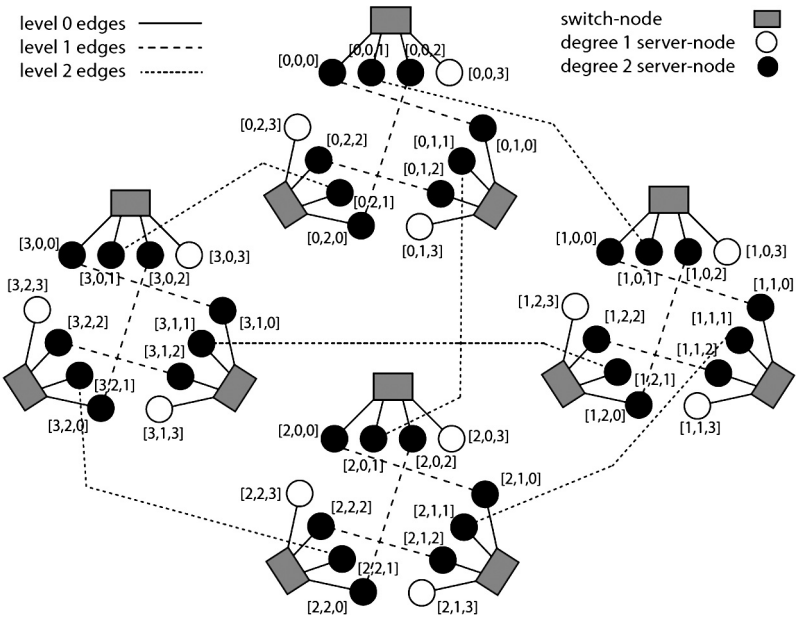

Fig. 5. A visualization of $\mathrm{FiConn}_{2,4}$.

Consequently, given k and n, it is but a simple calculation to use Theorem 14 to obtain $bw_S(\mathrm{GQ}^*_{k,n})$ for specific values of k and n; indeed, we do this in the next section.

We end this section with an important remark. Note that for a specific graph G we do not necessarily need to know extensive edge isoperimetric information regarding G, as per Theorem 14, in order to calculate $bw_S(G^*)$; for in Theorem 12 we have already calculated $bw_S(Q^*_k)$ using a combinatorial analysis specifically geared towards Q^*_k.

5. An empirical evaluation of GQ* and FiConn

Armed with our theory from the previous section, we can now empirically compare the S-bisection widths of GQ* and FiConn. Although this paper is primarily theoretical, it is useful to extend the analysis of GQ* against FiConn from [15] to consider S-bisection width. We begin by briefly describing the DCN FiConn before explaining our experimental set-up and our results.

5.1. The DCN FiConn

FiConn [31] is the best-known dual-port DCN and it serves as a reference point for comparing new dual-port designs. For any even $n \geq 2$, $\mathrm{FiConn}_{k,n}$ is a recursively-defined DCN where k denotes the level of the recursive construction and n the number of server-nodes that are directly connected to a switch-node (so, all switch-nodes have n ports). $\mathrm{FiConn}_{0,n}$ consists of n server nodes each of which is joined to a unique switch-node. Suppose that $\mathrm{FiConn}_{k,n}$ has b server-nodes of degree 1 (it can easily be verified that $b > 0$ is even; remember that n is always even). In order to build $\mathrm{FiConn}_{k+1,n}$, we take $\frac{b}{2} + 1$ copies of $\mathrm{FiConn}_{k,n}$ and for every copy, we choose $\frac{b}{2}$ server-nodes of degree 1 ensuring that each of these server-nodes is joined to a server-node of degree 1 in some other copy of $\mathrm{FiConn}_{k,n}$ so that every other copy of $\mathrm{FiConn}_{k,n}$ is represented (these additional links are called level $k+1$ links). The actual construction of which server-node is connected to which is detailed precisely in [31]; in particular, there is a well-defined naming scheme where server-nodes of $\mathrm{FiConn}_{k,n}$ are named as specific k-tuples of integers. In fact, although it is not made clear in [31], there is a multitude of connection schemes realising different versions of $\mathrm{FiConn}_{k,n}$. $\mathrm{FiConn}_{2,4}$, as constructed in [31], can be visualised in Fig. 5.

Some basic properties of $\mathrm{FiConn}_{k,n}$ are as follows: if we denote the number of server-nodes in $\mathrm{FiConn}_{k,n}$ by N_k then $N_0 = n$ and $N_{k+1} = N_k(\frac{N_k}{2^{k+1}} + 1)$; if we denote the number of switch-nodes in $\mathrm{FiConn}_{k,n}$ by E_k then $E_0 = 1$ and $E_{k+1} = E_k(\frac{N_k}{2^{k+1}} + 1)$; the number of level k links in $\mathrm{FiConn}_{k,n}$ is $\frac{N_{k-1}}{2^{k+1}}(\frac{N_{k-1}}{2^k} + 1)$; and the precise diameter of $\mathrm{FiConn}_{k,n}$ is unknown but known to be at most $2^{k+1} - 1$.

Whilst an exact value of the S-bisection width of $\mathrm{FiConn}_{k,n}$ is as yet unknown, a lower bound is given in [31] as $\frac{N_k}{2^{k+2}}$, where N_k is the number of server-nodes in $\mathrm{FiConn}_{k,n}$. In order to obtain an accurate comparison between the S-bisection widths of GQ* and FiConn, we must ensure that any lower bound we use is not overly conservative. To this end, we give an upper bound for $\mathrm{FiConn}_{k,n}$ that is not much larger than the lower bound from [31].

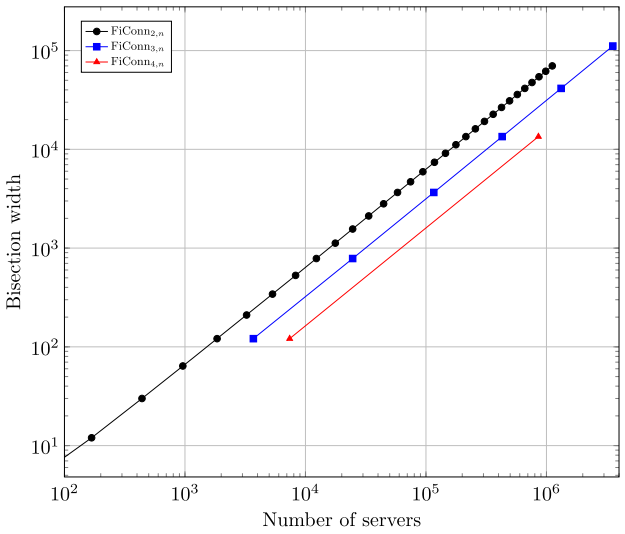

Fig. 6. An upper bound on $bw_S(\text{FiConn}_{k,n})$ with up to 4 million server-nodes and switch-nodes with up to 64 ports.

Theorem 16. *Let $b_{k,n} = \frac{N_k}{2^{k+2}}$ be the lower bound on the S-bisection width of $\text{FiConn}_{k,n}$ from [31] and let F_k be the number of canonical copies of $\text{FiConn}_{k-1,n}$ in $\text{FiConn}_{k,n}$. The S-bisection width of $\text{FiConn}_{k,n}$ is at most $(1+\frac{2^k}{N_{k-1}})b_{k,n}$, if F_k is even, and at most $(1+\frac{1}{F_k})b_{k,n}$, if F_k is odd.*

Proof. From [31], we have $F_k = \frac{N_{k-1}}{2^k} + 1$, for $k > 0$. If F_k is even then we can partition the set of copies into two sets X and Y so that each set contains $\frac{F_k}{2}$ copies of $\text{FiConn}_{k-1,n}$. What results is an S-bisection of $\text{FiConn}_{k,n}$ of width $\frac{F_k^2}{4}$.

Alternatively, if F_k is odd then let X comprise $\lfloor \frac{F_k}{2} \rfloor$ of the copies of $\text{FiConn}_{k-1,n}$ and let Y comprise the remaining $|X|+1$ copies. Denote the $|X|N_{k-1}$ server-nodes in X by S_X and the $(|X|+1)N_{k-1}$ server-nodes in Y by S_Y. This partition clearly does not induce an S-bisection of $\text{FiConn}_{k,n}$. However, we can amend the partition by moving server-nodes in S_Y to S_X. Suppose that (u, v) is a link joining a server-node $u \in S_X$ with a server-node $v \in S_Y$. Move v from S_Y to S_X. This yields a partition that is closer to an S-bisection of $\text{FiConn}_{k,n}$ but so that the width $\lfloor \frac{F_k}{2} \rfloor (\lfloor \frac{F_k}{2} \rfloor + 1) = \frac{F_k^2-1}{4}$ has not changed. In order to obtain an S-bisection we need to be able to undertake this amendment $\frac{N_{k-1}}{2}$ times; that is, we need that $\lfloor \frac{F_k}{2} \rfloor (\lfloor \frac{F_k}{2} \rfloor + 1) \geq \frac{N_{k-1}}{2}$. From above, this condition is the same as $\frac{N_{k-1}}{2^{k+1}}(\frac{N_{k-1}}{2^{k+1}} + 1) \geq \frac{N_{k-1}}{2}$, *i.e.*, $N_{k-1} \geq 2^{k+1}(2^k - 1)$. However, from [31], $N_{k-1} \geq 2^{k+1}(\frac{n}{4})^{2^k}$, when $n > 4$. As $(\frac{6}{4})^{2^k} > 2^k - 1$ whenever $k \geq 1$, we get that if $n \geq 6$ then we can obtain our S-bisection. The inequality can be easily verified by hand when $n = 4$.

So, if F_k is even (resp. odd) then we obtain an S-bisection of $\text{FiConn}_{k,n}$ of width $\frac{F_k^2}{4}$ (resp. $\frac{F_k^2-1}{4}$). By [31], $N_k = N_{k-1}(\frac{N_{k-1}}{2^k} + 1)$. Using this identity along with the lower bound $b_{k,n} = \frac{N_k}{2^{k+1}}$ and the identity $F_k = \frac{N_{k-1}}{2^k} + 1$ yields the result. □

For k small, the upper and lower bounds for the S-bisection width of FiConn are very close (as we shall soon see, we are actually only interested in $\text{FiConn}_{k,n}$ when $k = 2$ and $k = 3$). For example, when $k = 2$ and $n = 8$ (resp. $k = 2$ and $n = 48$) we have that $b_{2,8} = 28$ (resp. $b_{2,48} = 22{,}575$) and the upper bound from Theorem 16 is 30 (resp. 22,650). The upper bounds on the S-bisection width of $\text{FiConn}_{k,n}$ are plotted in Fig. 6 with $k = 2, 3, 4$ and $n \leq 64$ (so, there are up to around 4 million server-nodes). Note that the S-bisection width of $\text{FiConn}_{k,n}$ is smaller, by approximately a factor of 2, for $k = 3$ than it is for $k = 2$. This trend continues as k increases (this is not immediate from our graph). We remark that if we were to plot the lower bounds on Fig. 6 then they would appear almost identical to the upper bounds; so, for clarity, we have omitted them.

5.2. The DCN GQ*

The computational method underlying Theorem 14 allows us to compute the exact S-bisection width of the DCN GQ* where switch-nodes have radix up to 64 and where we have up to 4 million server-nodes (we use a simple Python script which takes about 1 minute on a standard laptop). In Fig. 7 we plot the S-bisection width against the number of servers for various DCNs $\text{GQ}^*_{k,n}$ (note that once we fix the number of server-nodes and n, the value of k is predetermined). For the sake of interest we also plot the S-bisection width of the stellar hypercube Q^*_k. We remark that because of its lower switch-radix relative to the number of server-nodes, the S-bisection width of Q^*_k is lower than that of the DCNs $G^*_{k,n}$ of comparable size.

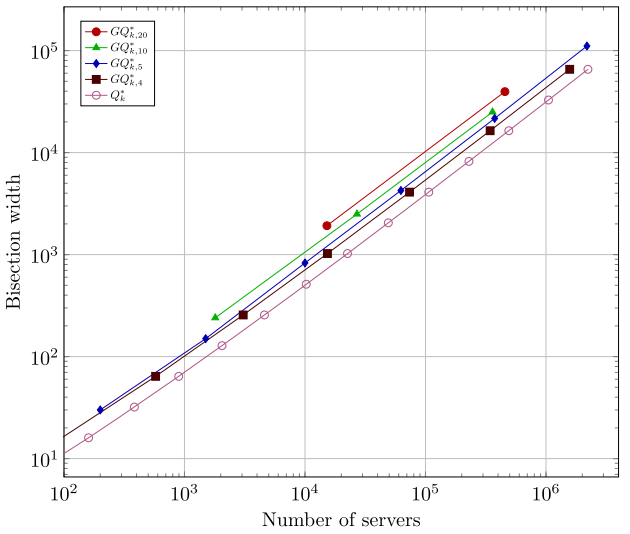

Fig. 7. The S-bisection widths of the DCN $\mathrm{GQ}^*_{k,n}$ with up to 4 million server-nodes and switch-nodes of radix up to 64.

In general, for a fixed number of server-nodes, the S-bisection widths of $\mathrm{GQ}^*_{k,n}$ plotted in Fig. 7 grow as the parameter n increases.

With reference to Fig. 7, there appears to be a rather predictable trend and perhaps there is a closed form to describe this data. However, we leave its derivation for future work.

5.3. Comparing GQ* and FiConn

In [15], GQ* and FiConn were empirically compared on the basis of network throughput, latency, load balancing, fault-tolerance, and cost-to-build (network throughput was measured according to the aggregate bottleneck throughput). Network throughput, load balancing, and fault tolerance were evaluated with respect to a routing algorithm *GQSRouting* for $\mathrm{GQ}^*_{k,n}$, based on well-known fault-tolerant routing algorithms for GQ (surveyed by Young and Yalamanchili in [43]). The experiments in [15] show that GQ* outperforms FiConn in all evaluations undertaken. Omitted from [15], however, is any empirical discussion of S-bisection width. We now compare the S-bisection widths of GQ* and FiConn.

The S-bisection width of a DCN can only be interpreted relative to other properties of the DCN; therefore, a meaningful comparison of the S-bisection widths of GQ* and FiConn can only be obtained when these other properties are also comparable. However, the question remains as to which other properties we should focus on. Comparing DCNs with a similar number of server-nodes, *i.e.*, size, is evidently desirable, but as adjusting the two parameters k and n can yield DCNs of similar size with starkly different properties, it does not make sense to simply normalise S-bisection width by network size. We focus on basic properties that can be made comparable by adjusting the parameters k and n of the networks in question, while maintaining an approximately constant network size. The properties we choose are the hardware costs and the diameter; that is, we compare $\mathrm{GQ}^*_{k',n'}$ with $\mathrm{FiConn}_{k,n}$ when both the hardware costs and the diameters are roughly the same.

First, we explain why we limit the parameter k in the DCNs $\mathrm{FiConn}_{k,n}$ that we examine to $k = 2$ and $k = 3$. FiConn is presented in [31] along with two routing algorithms: *traffic oblivious routing* (TOR); and *traffic aware routing* (TAR). The algorithm TOR is a straightforward recursive algorithm which is used as a building block for TAR, and the hop-lengths of the paths computed by TOR give the best known general upper bound for the diameter of $\mathrm{FiConn}_{k,n}$. What results is that $\mathrm{FiConn}_{k,n}$ has an effective diameter of $2^{k+1} - 1$, even if the true diameter is less than this, and thus FiConn becomes impractical to deploy for $k > 3$, since $\mathrm{FiConn}_{4,n}$ would have an effective diameter of at least 31 with currently known routing algorithms. Therefore, we only record our analysis for $k = 2$ and $k = 3$ but note that it holds for all small k.

Having explained how we only consider $\mathrm{FiConn}_{2,n}$ and $\mathrm{FiConn}_{3,n}$, we now explain how we ensure that the DCNs $\mathrm{GQ}^*_{k',n'}$ and $\mathrm{FiConn}_{k,n}$ (where $k = 2$ or $k = 3$) that are compared have similar hardware costs. We begin by showing that when G is a regular graph, the total hardware cost of G^* is proportional to the number of server-nodes, *i.e.*, the size, and that this holds for FiConn too.

We assume that the cost of a switch-node of some DCN with switch-nodes W and server-nodes S is proportional to its radix (this assumption is justified in [37] for switches of radix of up to around 100-150 ports); so, let the cost of a switch-node of W be C_p units per port and let the cost of a server-node of S be C_s units. Hence, when G is a regular graph of degree d, the total cost of the switch-nodes in G^* is $C_p d|W|$ and the total cost of the server-nodes is $C_s|S|$. As $|S| = d|W|$, we have that the total cost of the switch-nodes and server-nodes of G^* is $|S|(C_p + C_s)$. For FiConn, as each server-node is adjacent to exactly one switch-node, the number of switch-ports used is $|S|$. Hence, the total cost of the switch-nodes and

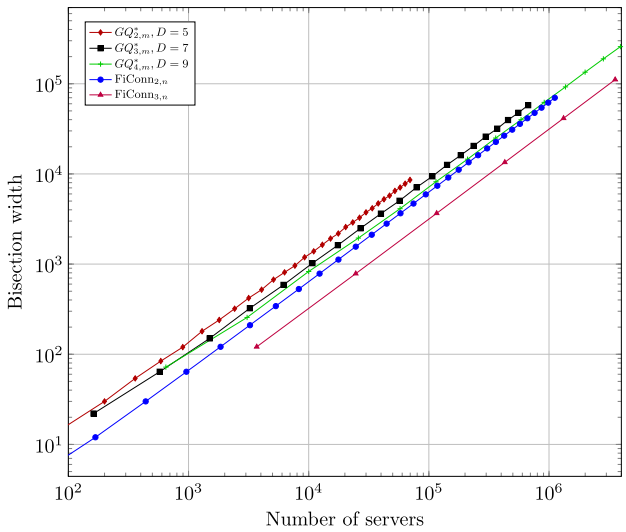

Fig. 8. The S-bisection widths of $\text{FiConn}_{2,n}$, $\text{FiConn}_{3,n}$, and $\text{GQ}^*_{k,n}$ of comparable sizes and diameters where there are up to 4 million server-nodes and switch-nodes of radix up to 64.

Table 1
Three DCNs with around 24 ports per switch.

DCN	$\text{GQ}^*_{3,10}$	$\text{GQ}^*_{4,6}$	$\text{FiConn}_{2,24}$
server-nodes	27,000	25,920	24,648
switch-nodes	1,000	1,296	1,027
switch-radix	27	20	24
links	81,000	77,760	67,782
diameter	7	9	7
bw_S	2,496	1,944	1,560

server-nodes of $\text{FiConn}_{k,n}$ is $|S|(C_p + C_s)$. However, if G^* and FiConn have the same number of server-nodes then FiConn has fewer server-node-to-server-node links than G^* does. In spite of this, the cost of the server-nodes and the switch-nodes is much greater than the cost of the server-node-to-server-node links and so we feel it is entirely reasonable to equate the hardware cost of either G^* or FiConn with the number of server-nodes.

Let us now consider relative diameters. Recall that the effective diameter of $\text{FiConn}_{k,n}$ is $2^{k+1} - 1$ and, from [15], the diameter of $\text{GQ}^*_{k,n}$ is $2k+1$, when $n > 2$, and $2k$, when $n = 2$. The DCNs $\text{GQ}^*_{2,24}$ and $\text{FiConn}_{3,8}$, for example, have 26,496 and 24,640 servers, respectively, but the S-bisection width of $\text{GQ}^*_{2,24}$ is over 4 times that of $\text{FiConn}_{3,8}$. However, the (effective) diameters of $\text{GQ}^*_{2,24}$ and $\text{FiConn}_{3,8}$, 5 and 15, respectively, differ so much that it is difficult to conceive of an application where one might replace the other. Hence, not only do we ensure that the numbers of server-nodes in two DCNs to be compared are roughly the same but we do likewise as regards their diameters.

Diameter is controlled in the plot of Fig. 8 as follows. First, for some fixed number N of server-nodes, we choose n so that $\text{FiConn}_{2,n}$ has roughly N server-nodes. We then calculate the (effective) diameter of $\text{FiConn}_{2,n}$ and we choose $\text{GQ}^*_{k,m}$ so that here are roughly N server-nodes and the diameter of $\text{GQ}^*_{k,m}$ is roughly that of $\text{FiConn}_{2,n}$ which is 7 (by 'roughly' we mean within 2). Note that in Fig. 8, D denotes the diameter of $\text{GQ}^*_{k,m}$ and we also include the S-bisection widths of $\text{FiConn}_{3,n}$; however, as the (effective) diameter of $\text{FiConn}_{3,n}$ is 15, we do not pursue comparisons with $\text{GQ}^*_{k,m}$.

5.4. Evaluation

Our results show that with respect to S-bisection width, the DCN GQ* outperforms the DCN FiConn when the hardware costs and the diameters are made (roughly) equal. This accentuates the advantages of GQ* over FiConn as derived in [15]. We have pulled out a typical comparison in Table 1. The hardware costs of all DCNs are comparable as are their diameters. However, there are significant differences in their S-bisection widths, with even $\text{GQ}^*_{4,6}$ performing better than $\text{FiConn}_{2,24}$ despite having higher diameter and lower radix switch-nodes. In general, DCNs of the form $\text{GQ}^*_{3,n}$ have S-bisection widths of roughly 1.5 times those of DCNs of the form $\text{FiConn}_{2,n}$ when hardware costs are levelled (all of these DCNs have diameter 7). If we consider DCNs of the form $\text{GQ}^*_{4,n}$ (so that the diameter increases to 9) then the S-bisection widths are still comparable with DCNs of the form $\text{FiConn}_{2,n}$, and the DCNs of the form $\text{GQ}^*_{4,n}$ host more server-nodes.

6. Conclusions

In this paper we have achieved a number of novel research results: we have justified the use of bisection width, or more precisely S-bisection width, as a valid metric in the evaluation of structured and symmetric server-centric DCNs; we have shown how the study of isoperimetric problems in graph theory is highly relevant to the computation of the S-bisection width of dual-port server-centric DCNs built according to a widely-applicable stellar methodology; we have precisely evaluated the S-bisection width of stellar hypercubes; we have obtained a general method (albeit highly dependent on research into isoperimetric problems) for computing the S-bisection width of stellar DCNs when the base graph is regular; and we have applied our methods in order to show that when compared with FiConn, stellar generalized hypercubes have much better properties with respect to S-bisection width (as they do as regards aggregate bottleneck throughput, latency, load balancing, fault-tolerance, and cost-to-build).

We note that our exploitation of discrete mathematics within DCN design is not unique to this paper: in [39] we used existing research on WK-recursive interconnection networks to develop improved routing algorithms for the DCNs HCN and BCN; in [13] we used the recursive structure of the DCNs DCell and FiConn to develop new, improved proxy-routing algorithms (our techniques can be applied more widely); in [40] combinatorial design theory was used to build switch-centric DCNs that compare favourably with Fat-Trees in terms of fault-tolerance; and in [14] we proved that an abstraction of DPillar is a Cayley graph and used the consequent node-symmetry to devise an optimal routing algorithm. In short, we are showing that discrete mathematics has a fundamental role to play as regards the design of DCNs.

There are many directions for further research. The primary one is the continued study of the general stellar mechanism; that is, we should be examining other base graphs G in tandem with the properties of G^* and in comparison with other dual port server-centric DCNs. Naturally, this study should pay attention to what has been proven as regards isoperimetric problems (and, indeed, should inspire more work on such problems). Moreover, this study should involve metrics which have so far not been studied and it should also consider the generic utilization of stellar DCNs within cloud computing; for example, with regard to the capacity of these DCNs for virtualization.

Also, we intend to examine our generic stellar construction against similar ones where edge subdivision is done by introducing one server-node (as was done to obtain the DCN SWCube from generalized hypercubes in [32]) and also three or more server-nodes. On the face of it, our stellar construction should be preferable: we can incorporate more server-nodes than if we subdivide edges with one server-node; and we retain symmetry unlike when we subdivide edges using three or more server-nodes. This examination will be both analytical and empirical.

Finally, associated with the further investigation of the stellar methodology in the dual-port server-centric environment are generalizations so as to convert graphs into multi-port DCNs. Of course, such generalizations would attempt to utilize beneficial properties of the base graph. We say no more about this direction of research here beyond that it will involve more complex graph substitution mechanisms, *e.g.*, replacing edges with networks of server-nodes and possibly introducing additional links between server-nodes so introduced (but in a structured and controllable manner).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, A. Donnelly, Symbiotic routing in future data centers, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (August 2010) 51–62.
[2] J.H. Ahn, N. Binkert, A. Davis, M. McLaren, R.S. Schreiber, HyperX: topology, routing, and packaging of efficient large-scale networks, in: Proc. of Conf. on High Performance Computing Networking, Storage and Analysis, ACM, 2009, page article no. 41.
[3] M. Al-Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (October 2008) 63–74.
[4] J.A. Aroca, A.F. Anta, Bisection (band)width of product networks with application to data centers, IEEE Trans. Parallel Distrib. Syst. 25 (3) (March 2014) 570–580.
[5] M. Besta, T. Hoefler, Slim fly: a cost effective low-diameter network topology, in: Proc. of Int. Conf. for High Performance Computing, Networking, Storage and Analysis, IEEE, 2014, pp. 348–359.
[6] S. Bezrukov, Edge isoperimetric problems on graphs, in: L. Lovász, A. Gyarfas, G.O.H. Katona, A. Recski, L. Szekely (Eds.), Graph Theory and Combinatorial Biology, in: Bolyai Society Mathematical Studies, vol. 7, Janos Bolyai Mathematical Society, 1999, pp. 157–197.
[7] S.N. Bhatt, F.T. Leighton, A framework for solving VLSI graph layout problems, J. Comput. Syst. Sci. 28 (2) (April 1984) 300–343.
[8] L.N. Bhuyan, D.P. Agrawal, Generalized hypercube and hyperbus structures for a computer network, IEEE Trans. Comput. C-33 (4) (April 1984) 323–333.
[9] T.N. Bui, S. Chaudhuri, F.T. Leighton, M. Sipser, Graph bisection algorithm with good average case behaviour, Combinatorica 7 (2) (June 1987) 171–191.
[10] W. Dally, B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2003.
[11] J. Díaz, J. Petit, M.J. Serna, A survey of graph layout problems, ACM Comput. Surv. 34 (3) (September 2002) 313–356.
[12] R. Diestel, Graph Theory, Graduate Texts in Mathematics., vol. 173, Springer, 2012.
[13] A. Erickson, A. Kiasari, J. Navaridas, I.A. Stewart, Routing algorithms for recursively-defined data centre networks, in: Proc. of 13th IEEE Int. Symp. on Parallel and Distributed Processing With Applications, IEEE, 2015, pp. 84–91.
[14] A. Erickson, A. Kiasari, J. Navaridas, I.A. Stewart, An optimal single-path routing algorithm in the datacenter network DPillar, IEEE Trans. Parallel Distrib. Syst. 28 (3) (March 2017) 689–703.
[15] A. Erickson, I.A. Stewart, J. Navaridas, A.E. Kiasari, The stellar transformation: from interconnection networks to datacenter networks, Comput. Netw. 113 (Feb. 2017) 29–45.

[16] N. Farrington, G. Porter, S. Radhakrishnan, H.H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, A. Vahdat, Helios: a hybrid electrical/optical switch architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (October 2010) 339–350.
[17] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.
[18] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, S. Sengupta, VL2: a scalable and flexible data center network, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (August 2009) 51–62.
[19] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, S. Lu, BCube: a high performance, server-centric network architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (August 2009) 63–74.
[20] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, S. Lu, DCell: a scalable and fault-tolerant network structure for data centers, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (August 2008) 75–86.
[21] D. Guo, T. Chen, D. Li, M. Li, Y. Liu, G. Chen, Expandable and cost-effective network structures for data centers using dual-port servers, IEEE Trans. Comput. 62 (7) (July 2013) 1303–1317.
[22] D. Guo, C. Li, J. Wu, X. Zhou, DCube: a family of network structures for containerized data centers using dual-port servers, Comput. Commun. 53 (November 2014) 13–25.
[23] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S.R. Das, J.P. Longtin, H. Shah, A. Tanwer, Firefly: a reconfigurable wireless data center fabric using free-space optics, ACM SIGCOMM Comput. Commun. Rev. 44 (4) (October 2014) 319–330.
[24] L.H. Harper, Optimal assignments of numbers to vertices, J. Soc. Ind. Appl. Math. 12 (1) (March 1964) 131–135.
[25] S.A. Jyothi, A. Singla, P.B. Godfrey, A. Kolla, Measuring and Understanding Throughput of Network Topologies, Proc. of Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC'16), vol. 65, IEEE, 2016.
[26] J. Kim, W.J. Dally, D. Abts, Flattened butterfly: a cost-efficient topology for high-radix networks, ACM SIGARCH Comput. Archit. News 35 (2) (June 2007) 126–137.
[27] J. Kim, W.J. Dally, S. Scott, D. Abts, Technology-driven, highly-scalable dragonfly topology, in: Proc. of 35th Ann. Int. Symp. on Computer Architecture, IEEE, 2008, pp. 77–88.
[28] M. Kliegl, J. Lee, J. Li, X. Zhang, C. Guo, D. Rincon, Generalized DCell structure for load-balanced data center networks, in: Proc. of IEEE INFOCOM Conf. on Computer Communications Workshops, IEEE, 2010, pp. 1–5.
[29] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Array, Trees, Hypercubes, Morgan Kaufmann, 1992.
[30] C.E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE Trans. Comput. 34 (10) (October 1985) 892–901.
[31] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, J. Wu, Scalable and cost-effective interconnection of data-center servers using dual server ports, IEEE/ACM Trans. Netw. 19 (1) (February 2011) 102–114.
[32] D. Li, J. Wu, On the design and analysis of data center network architectures for interconnecting dual-port servers, in: Proc. of IEEE INFOCOM, IEEE, 2014, pp. 1851–1859.
[33] Y. Liao, J. Yin, D. Yin, L. Gao, DPillar: dual-port server interconnection network for large scale data centers, Comput. Netw. 56 (8) (May 2012) 2132–2147.
[34] J.H. Lindsey, Assignment of numbers to vertices, Am. Math. Mon. 71 (5) (May 1964) 508–516.
[35] B. Monien, R. Preis, S.S. Schamberger, Approximation algorithms for multilevel graph partitioning, in: T.F. Gonzalez (Ed.), Handbook of Approximation Algorithms and Metaheuristics, Chapman and Hall/CRC, 2007, pp. 60.1–60.15.
[36] K. Nakano, Linear layouts of generalized hypercubes, in: Proc. of Graph-Theoretic Concepts in Computer Science, in: Lecture Notes in Computer Science, vol. 790, Springer, 1994, pp. 364–375.
[37] L. Popa, S. Ratnaswamy, G. Iannaccone, A. Krishnamurthy, I. Stoica, A cost comparison of data center network architectures, in: Proc. of 6th Int. Conf. on Emerging Networking Experiments and Technologies, vol. 16, ACM, 2010.
[38] A. Singla, C.-Y. Hong, L. Popa, P.B. Godfrey, Jellyfish: networking data centers randomly, in: Proc. of 9th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, 2012, pp. 225–238.
[39] I.A. Stewart, Improved routing in the data centre networks HCN and BCN, in: Proc. of 2nd Int. Symp. on Computing and Networking - Across Practical Development and Theoretical Research, IEEE, 2014, pp. 212–218.
[40] I.A. Stewart, On the mathematics of data centre network topologies, in: Proc. of 20th Int. Symp. on Fundamentals of Computation Theory, in: Lecture Notes in Computer Science, vol. 9210, Springer, 2015, pp. 283–295.
[41] C.D. Thompson, A Complexity Theory for VLSI, PhD thesis, Carnegie-Mellon University, 1980.
[42] R.V. Tomic, Optimal networks from error correcting codes, in: Proc. of ACM/IEEE Symp. on Architectures for Networking and Communications Systems, ACM, 2013, pp. 169–179.
[43] S.D. Young, S. Yalamanchili, Adaptive routing in generalized hypercube architectures, in: Proc. of 3rd IEEE Symp. on Parallel and Distributed Processing, IEEE, 1991, pp. 564–571.